

```
def rectangle_perimeter(r0, c0, width, height, shape=None, clip=False):
    rr, cc = [r0, r0 + width, r0 + width, r0], [c0, c0, c0 + height, c0 + height]
    return skimage.draw.polygon_perimeter(rr, cc, shape=shape, clip=clip)
# functie om bounding boxes te tekenen
```

```
start_size = round(im_height/10) # startgrootte van het window
stepsize = 40 # per hoeveel pixels het window verschuift
detectedfaces = []
image_detected = image.copy()
```

```
for w in range(start_size,int(im_width/4),20): # for loop voor verschillende window groottes
    if w <= im_height and w<= im_width: # check of window niet buiten de image gaat
        for y in range(0,im_height-w, stepsize): # for-loop om vertikaal te schuiven
            for x in range(0,im_width-w, stepsize): # for-loop om horizontaal te schuiven
                im_w = image[y:y+w,x:x+w,:] # knip het stukje van de afbeelding uit dat onder dat
#zich onder het window bevindt.
                im_in = transform.resize(im_w,(64, 64),mode='constant',anti_aliasing=True)# het
neuraal netwerk verwacht hier een image van 64 bij 64 pixels. Dit kan bij jouw code anders
zijn.
                im_in = np.expand_dims(im_in, axis=0)
                face = model.predict_proba(im_in) #predictie
                if face[0,1] > 0.98: # manuele threshold om het model gevoeliger of minder
gevoelig te maken voor gezichten
                    print('face detected:', model.predict_proba(im_in))

                rr, cc = rectangle_perimeter(y, x, w, w) # tekenen van een bounding box rond het
gezicht
                image_detected[rr,cc] =255
```